

## THREADED PRESENTATION OF ELECTRONIC MAIL

### Inventors

Scott Forstall  
Greg Christie  
Stephen O. Lemay

## COPYRIGHT NOTICE

[0001] A portion of this disclosure contains material in which copyright is claimed by the applicant. The applicant does not object to the copying of this material in the course of making copies of the application file or any patents that may issue on the application, but all other rights whatsoever in the copyrighted material are reserved.

## BACKGROUND OF THE INVENTION

### Field of the Invention

[0002] The present invention relates generally to the field of presentation of electronic mail. More specifically, the present invention provides an improved access to e-mail display by using threading to organize messages in a user's mailbox.

### Description of the Related Art

[0003] As the amount of e-mail people receive continues to rise almost geometrically, there is an increased need for a way in which to manage that e-mail. When a user receives dozens or even hundreds of e-mails in a single day, the messages become difficult to usefully organize and contextualize. Add to that the occasional vacation, leaving no access to e-mail, and imagine the horror waiting for the user upon his return to find several hundred unread e-mails.

[0004] In an effort to help users get a handle on displaying their e-mails, some solutions have already been tried. The most simple example allows a user's inbox to be sorted by one or more various criteria. For example, a user can sort his inbox by

message sender, by subject, by date, by file size, or by conversation. Sorting by message sender is helpful because it allows all messages from a single sender to be grouped together in the inbox view. This approach ignores the fact that individual e-mail messages are often part of a chain of messages from different users, forming a discussion. The user therefore loses the context of the conversation. That is, one can see only one side of the conversation. If someone else, including the mailbox user himself has responded to the sender in between two of the sender's messages, that response is lost. Thus, sorting messages merely by message sender is not an adequate solution to the problem of organizing one's inbox. Similarly, organizing messages by received date or by sent date is useful in that it allows messages to be read in the order in which they were received or sent, but does not really give the mailbox owner any context, especially since the user may be participating in multiple conversations at once via e-mail, and there is no grouping of the individual conversations. This problem is addressed partially by the ability to group messages by "conversation", offered by Outlook, a product of Microsoft Corporation of Redmond, Washington. Outlook simply lists messages that are related to a single conversation, even if any message sender subsequently changes the subject line of the message in response to a previous message in that conversation. While this partially addresses the problem, the grouping adds no information about the group itself—it merely organizes the messages to be together in a list in the user's view of the inbox. In addition, all messages become conversations, even where there is only a single message in the conversation. This only adds extra overhead to the already encumbered inbox view. Thus, there is still room for refinement.

**[0005]** In view of the foregoing, a need therefore exists for a way to manage a user's inbox that organizes e-mails in such a way as to provide a thread of messages, including contextual information about the contents of each thread.

## SUMMARY OF THE INVENTION

**[0006]** The present invention satisfies the foregoing need by providing a threaded list of messages in a user's mailbox. Each message received is checked to determine whether the new message is associated with one or more previous messages. If the received message is related to a single previous message, then a new thread is created. The new thread has an associated thread header, which stores metadata about the messages in the thread. If the received message is related to messages in an existing thread, then the received message is added to that thread, and the thread header for that thread is updated to reflect data from the received message. The thread headers are persistent data, that are stored in addition to the actual messages themselves. This allows various types of organizational and viewing operations to be performed directly on the thread headers. In a preferred embodiment, the thread header appears along with messages in a message list view of the user's inbox, and provides data about the thread to the user, including the number of messages in the thread, originator of the thread, the subject of the thread, and the time that the thread was most recently updated. Messages that are not part of a thread are preferably shown in the same view as threaded messages, but without a thread header. This allows a user to view both threaded and non-threaded messages in a single view.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0007]** Fig. 1 is a block diagram of a system in accordance with an embodiment of the present invention.

**[0008]** Fig. 2 illustrates a view of the user's mailbox prior to activating the threading feature in accordance with an embodiment of the present invention.

**[0009]** Figs. 3 and 4 illustrate the selection of an "Organize by Thread" option in accordance with an embodiment of the present invention.

[0010] Fig. 5 illustrates an example view of a user's inbox with threading turned on in accordance with an embodiment of the present invention.

[0011] Fig. 6 illustrates a view of a collapsed thread header in a user's inbox in accordance with an embodiment of the present invention.

[0012] Fig. 7 illustrates a view of a user's inbox in which a selected thread header is located towards the top of the message index list in accordance with an embodiment of the present invention.

[0013] Fig. 8 illustrates a view of the user's inbox in which a thread header has been expanded in accordance with an embodiment of the present invention.

[0014] Fig. 9 is a flowchart illustrating a method of threading e-mail in accordance with an embodiment of the present invention.

[0015] The figures depict preferred embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0016] Referring now to Fig. 1, there is shown a block diagram of a system in accordance with an embodiment of the present invention. System 100 includes a mailbox processor module 102 for receiving e-mails; threading engine 104 for organizing and displaying e-mail in a threaded fashion; and mailbox 106 for storing e-mail messages. System 100 can be implemented as part of a general e-mail and/or calendaring and/or contact management solution in a variety of ways, as will be appreciated by those of skill in the art. The system can operate on a user's client

computer, on a server computer, or in some combination using both such computers.

Those of skill in the art appreciate that the particular implementation and distribution of functionality between various computers is not material to the invention.

**[0017]** An e-mail message (or, simply, "an e-mail") existing on its own is not part of a thread. A thread exists when there are two or more e-mails that bear a relationship to one another. In a preferred embodiment, an e-mail is part of a thread if it is a reply to an earlier message, or a forward of an earlier message. The earlier message is also part of the thread, once the thread comes into being. Note that the e-mails in a thread do not have to be located within a single folder in the user's mailbox. For example, a thread may include messages from the user's sent items folder, personal archive folders, inbox, and other folders used to organize the user's email. Accordingly, though the present invention is described in the context of threads within a single folder, it should be understood that this is for clarity of explanation only, and by no means intended as a limitation of the invention.

**[0018]** One way to determine whether an e-mail is part of a thread is to check whether messages have the same subject. In a preferred embodiment, when a message arrives, threading engine 104 first ignores any subject prefixes (such as RE:, FWD:, etc.) and then compares the remainder of the subject line heading with the subject headings of messages already in the recipient's inbox, here too ignoring their subject line prefixes. Because this method alone is over inclusive (two or more unrelated e-mails can have the same subject), a check is also performed to validate that the message rightfully belongs to the thread. For example, many people send e-mails having the subject "Hello." As a result, there may be many messages in the user's inbox that have the subject "Hello," but are not related to one another. Accordingly, in one embodiment, threading engine 104 checks the "In-Reply-To" field of the header of the incoming message to confirm that it shares the same message ID as an earlier message in the thread. The In-Reply-To

SMTP header is a standard field in Internet e-mail as specified by RFC 822. In an alternative embodiment, other headers may be used, such as the message ID header.

**[0019]** When it is determined that a thread exists, threading engine 104 generates a thread header for the thread. A thread header contains information about the thread that it heads, including in one embodiment a sender, a subject, a date, and a number of messages unread in the thread. The thread header is a set of data about the thread that exists independently of the thread itself. Each thread has its own thread header.

Threads within mailbox 106 can be sorted and displayed according to any sort criteria found in their respective thread header. For example, if the mailbox is sorted by Subject, then threads will be displayed in ascending or descending lexicographic order according to the subject of the earliest (oldest) message in the thread. If the mailbox is sorted by date, then threads will be displayed in order based upon the date appearing in the thread header, as set forth below.

**[0020]** Fig. 2 illustrates a view of the user's mailbox prior to activating the threading feature. In the illustrated embodiment of Fig. 2, the main view is split into two sub parts: a list 220 of email messages, and a content view 218 of the contents of either a message or thread selected in the list 220. Message list 220 includes a "From" column 202, a Subject column 204, and a Date Received column 206. Each of these columns is one that can be sorted on, and in the illustrated embodiment, Date Received 206 is the column being sorted on. Accordingly, each of the messages listed is sorted in the order in which it was received. A message that is currently being viewed is highlighted in list 220 in a preferred embodiment, and displayed in content view 218. In Fig. 2, message 208 is being viewed, and is from Scott Forstall, with the subject "Sampler Icon", and was received Wednesday, June 18, 2003 at 11:01 PM. The text of message 208 is displayed in content view 218.

[0021] Although messages in this view are not threaded, it is apparent from a review of inbox window 220 that some of the other messages are related to message 208. For example, messages 210, 212, 214, and 216 each have the subject "Sampler Icon." Accordingly, in a threaded view, we would expect to see these messages grouped together.

[0022] Fig. 3 shows how an option 304 can be selected from a menu such as View menu 302 to turn on "Organize by Thread." Fig. 4 illustrates that in one embodiment when the Organize by Thread option 402 is selected, a check mark appears next to the item in the menu 302.

[0023] Referring now to Fig. D there is shown an example view of a user's inbox with threading turned on. Again, the view is split into two panels: the message listing 520, and the content view 518 of a single message. In message listing 520, there are again three columns: a "From" column 502, "Subject" column 504, and "Date Received" column 506. In the illustrated embodiment, sorting is again being done on the Date Received 506 column. When threading is turned on, a thread column 522 is also displayed in a preferred embodiment. The presence of an arrow in the thread column 522 pointing either to the right or down indicates the beginning of a thread, and the line on which the arrow appears contains data from the thread header for the thread. Header 524 is one example of a thread header. Like a message line, header 524 includes From, Subject and Date Received data. Since the thread header is intended to give information to the user about the contents of the thread, contextual information from the thread is included in the header 524. For example, the From entry d28 provides the name of the person who started the thread. The Subject 530 in the thread header 524 is the original subject of the thread. The Date Received 532 entry is the date on which the most recent message in the thread was received. In alternative embodiments, other criteria could be used for the thread header. For example, the From entry could reflect

the name of the person who most recently sent a message in the thread, instead of the person who originated the thread. Similarly, the subject entry could be the subject of the message as changed by one of the forwarders or respondents in the thread, or it could be the originating subject; and the Date Received could be the date on which the first message was received in the thread, instead of the most recent message. Other modifications could also be made as will be readily appreciated by those of skill in the art. Appropriately, the five messages 208, 210, 212, 214 and 216 (Fig. 2) referred to above as belonging to a single conversation have now been grouped into a single thread 534. Each of the messages in the thread has the subject "Sampler Icon," and some of the subjects are preceded by subject prefixes such as "Re:" and "Fwd." The most recent message 516 was received Thursday June 19, 2003 at 12:38 AM, from Scott Forstall. This information is extracted and appears again in the thread header 524. Thread header 524 also includes a number of unread messages d26. In the case of the illustrated header 524, there are four unread messages in the thread. In a preferred embodiment, thread headers can be either expanded or collapsed. When a thread is collapsed, the arrow in column 522 points to the right, as is standard to indicate collapsed data. An example is the thread 536 from Ali Ozer. When expanded, the arrow points down, and the messages within the expanded thread are shown with the From data indented to help the user identify the messages as belonging to the thread. Using threads only where there are two or more related emails avoids the unnecessary clutter that would otherwise result if a single email was designated as a thread: if a thread header, such as header 524 was displayed for each individual message, then the user would expand the header only to find a single email. This only wastes the user's time, as there is no more useful information in the thread header than in the single email itself.

**[0024]** In one embodiment messages within a thread are displayed against a colored background to separate them from other threads, or from messages not within any



thread. For example, message 538 from Heather M. Snow with Subject "Ice cream today at 4pm" is not within a thread. It is preferably displayed against a white or plain background. On the other hand, messages within thread 534 are displayed against a light blue background; thread header 524 is displayed against a darker background; and the message currently being viewed 508 is displayed against a high contrast background. Those of skill in the art will recognize that the particular colors or shades of colors used are not material to the present invention, and that any suitable manner of demarcating threads can be used.

[0025] Fig. 6 shows thread header 524 in a collapsed mode, where only the header itself is visible in the message list area 602, and none of the messages within the thread is visible. However, in a preferred embodiment when the collapsed header is selected in the message listing area 602, content view area 604 shows the details of the thread header 606 and a list 608 of all the messages in that thread. Details of the thread header 606 include in a preferred embodiment the title of the thread, which is the original subject of the thread, the originator of the thread, and the time at which the thread was started, and the number of messages and unread messages in the thread. List 608 indicates for each of the messages in the thread its status (read, unread, attachments, dates, times, subjects, etc.). Selecting one of the messages from the list 608 expands the thread in the messages listing area 602 and displays the message that was selected.

[0026] Fig. 7 illustrates a view in which the selected thread header 702 is located towards the top of the message index list 704. When this thread is expanded to reveal its contents, highlighted bar 706 remains in the same position relative to the index screen 704, and the thread expands above and below the bar, showing the emails in the thread that are prior to the selected email (based on the current sort order) above the selected email, and those emails in the thread that are after the selected email below it. This preserves the visual context for the user by keeping the highlighted bar 706 consistently

located in the same position in the view. This is illustrated further with reference to Fig. 8, in which the thread has been expanded to reveal a message from Marcel van Os 808 located in the same position as the thread header 702 was previously located. In one embodiment, when messages belonging to a single thread are located in different folders, opening the thread header in any one folder brings all the messages together to be viewed. Collapsing the thread restores the messages to their original location.

[0027] When a new message arrives, threading engine 104 examines the subject and In-Reply-To fields of the message header to determine whether the message is related to any other messages already in the user's mailbox. Alternatively, the threading engine 104 may use just the In-Reply-To field to determine the existence of a thread. If the message is so related, and if a thread header has previously been created (i.e. there are already at least two related messages in the user's mailbox), the current message is added to the existing thread. Alternatively, if the message is related to another message, but a thread does not yet exist, a thread header is created for the new thread. The thread header information is updated, for example, to store the sender information for the most recent message, an updated count of the messages in the thread, the time of the most recent message, and any other attributes or fields that may be usefully included in the summary information of the thread header. The message inbox view is then adjusted as necessary, depending on its sort criteria. For example, if the message inbox is sorted according to Date Received, and the thread header indicates the date received of the most recent message, then when a new message arrives, the thread containing the new message will become the first message thread in the message list, if the list is in descending order of date received. This is illustrated further with respect to Fig. 9, in which a message has just arrived in the "remaining items for scott's WWDC slides" thread. Accordingly, that thread 904 is moved to the top of the message inbox list 902, since the date received category is being sorted on in descending order.

[0028] In one embodiment, when a user expands a thread, the oldest unread message in the expanded thread is automatically highlighted. One implication of this is that if a user reads a message, collapses the thread, and then re-expands the thread, he is automatically shown the next unread message in the thread. This enables the user to review a some of the messages in a thread, close the thread, and then return to it later, and immediately pick up the review where he left off, without having to remember which was the last message reviewed.

[0029] In one embodiment, mailbox processor module 104 supports attribute-based color coding of messages, in which messages having certain attributes are automatically colored a certain color. In particular, one attribute for color coding is the message sender, so that all messages from a given sender have the same color. When threading is activated, the color of a thread takes on the color, if any, of the sender who initiated the thread. In an alternative embodiment, the color of the thread is the color associated with the sender who sent the most recent e-mail in the thread. This allows the user to tell from the thread header who has most recently added a message to the thread. In another embodiment, each sender may have an associated priority value. In this case, the color of the thread will be the color of the sender of a message in the thread having the highest priority.

[0030] Referring now to Fig. 10, there is shown a flowchart of a method in accordance with an embodiment of the present invention. When an e-mail is received 1002 by mailbox processor module 102, threading engine 104 first determines 1004 whether the e-mail is related to either an existing thread or an earlier single message. If the message is not related to an existing thread or earlier message, no more threading is necessary for that message, and it is displayed 1006 in an unthreaded fashion in the mailbox index. If the message is part 1005 of an existing thread, then threading engine 104 associates 1014 the message with the existing thread. If the message is related to an

earlier single message but there is no existing thread, threading engine 104 creates 1010 a new thread header for the new thread, and associates 1012 both the earlier message and the current message with the new thread. Next, threading engine 104 updates 1016 the thread header with information gleaned from the incoming message. For example, in a preferred embodiment, the thread header is updated to have as its Date Received information the Date Received of the most recent e-mail in the thread. Finally, the message is displayed 1018 in the thread.

[0031] The present invention has been described in particular detail with respect to a limited number of embodiments. Those of skill in the art will appreciate that the invention may additionally be practiced in other embodiments. First, the particular naming of the components, capitalization of terms, the attributes, data structures, or any other programming or structural aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names, formats, or protocols. Further, the system may be implemented via a combination of hardware and software, as described, or entirely in hardware elements. Also, the particular division of functionality between the various system components described herein is merely exemplary, and not mandatory; functions performed by a single system component may instead be performed by multiple components, and functions performed by multiple components may instead performed by a single component. For example, the particular functions of the threading engine 104 and so forth may be provided in many or one module.

[0032] Some portions of the above description present the feature of the present invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are the means used by those skilled in the e-mail management arts to most effectively convey the substance of their work to others skilled in the art. These operations, while described functionally or

logically, are understood to be implemented by computer programs. Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules or code devices, without loss of generality.

**[0033]** It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the present discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices.

**[0034]** Certain aspects of the present invention include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and instructions of the present invention could be embodied in software, firmware or hardware, and when embodied in software, could be downloaded to reside on and be operated from different platforms used by real time network operating systems.

**[0035]** The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs,

magnetic or optical cards, application specific integrated circuits (ASICs), or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

**[0036]** The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description above. In addition, the present invention is not described with reference to any particular programming language. It is appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references to specific languages are provided for disclosure of enablement and best mode of the present invention.

**[0037]** Finally, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention.